

# Setting up the Magic Leap Development Environment and Hello Cube Tutorial



Adapted From:

Magic Leap Workshops Document

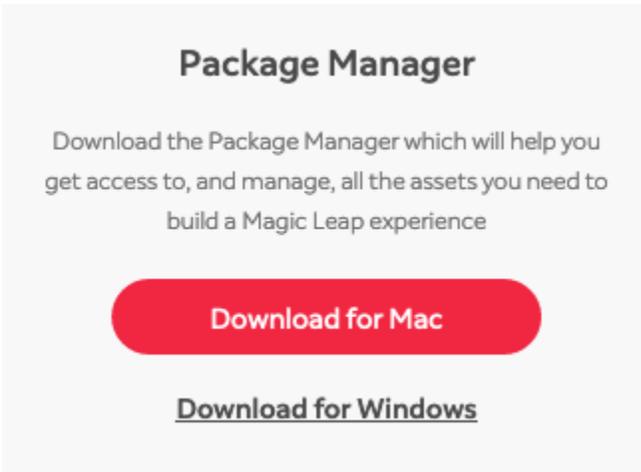
*Authors (Gemma Henderson, Siobhan Williams, Hammam Alsafrijalani, Chris Chung)*

And from:

Ching-Hua Chuan's [online blog](#) on Unity & Magic Leap

## Step 1: Install Magic Leap Package Manager

Go to <https://creator.magicleap.com/downloads> and download the Package Manager Installer. Run the Package Manager Installer and complete the setup wizard.



Install the following packages:

- From Common Packages:
  - Lumin SDK
  - C API Documentation
  - Visual Studio 2017 Extension (or VS Code extension if using VS Code)
- From Lumin Runtime Packages:
  - Lumin Runtime SDK
  - Lumin Runtime Documentation
- From Unity Packages:
  - Magic Leap Unity Package
  - Unity API Documentation

Magic Leap PackageManager

ALL PACKAGES

Shing Cheng

### Common Packages

- Lumin SDK 0.20.0 PENDING INSTALL
- C API Documentation 0.20.0 PENDING INSTALL
- Power and Thermal Profiler NOT INSTALLED
- NVIDIA Nsight Graphics NOT INSTALLED
- NVIDIA Nsight Systems NOT INSTALLED
- Visual Studio Code Extension NOT INSTALLED
- Visual Studio 2017 Extension 1.0.190301 PENDING INSTALL

### Change List

Package	Version	Size	Status
Lumin SDK	0.20.0	0.47 GB	✓
C API Documentation	0.20.0	3.8 MB	✓
Visual Studio 2017 Extension	1.0.190301	0.83 MB	✓

APPLY CANCEL

### Visual Studio 2017 Extension (1.0.190301)

Available Versions

Open the folder with the Magic Leap Extension for Visual Studio 2017 in a file browser

Launch the Magic Leap Extension for Visual Studio 2017 Installer

Open Folder Launch Installer

### DEPENDENCIES

- Lumin SDK (any version)

### Magic Leap Extension for Visual Studio 2017

This extension adds Magic Leap-specific functions to all Microsoft Windows editions (Community, Professional, or Enterprise) of Visual Studio 2017.

- Create, build and debug Magic Leap applications on device.
- Create, build and debug Magic Leap applications on host with simulator.
- Debug Magic Leap Unreal project on device.

Install Visual Studio 2017 before installing this extension package.

To install, click **Launch Installer** button, and then follow the Visual Studio Installation steps.

To uninstall the extension in Visual Studio, go to Visual Studio menu **Tools > Extensions and Updates**. Remove the installer binary by uninstalling the extension with Package Manager.

### What's New

- On creating/importing a project, source files from all components (including any library projects) referenced by underlying mabu project are included in the resulting VS project. Older versions only include source files from main mabu project.
- New debugger option **GDB command file** is added for user to specify a GDB command file for on-device debug.
- Show **IntelliSense** property page in Project Properties dialog.
- Added **"Browse..."** button for SDK option in Project Properties dialog.
- Greatly speed up project build by enabling parallel compiling. This applies to newly created projects. For existing projects, just add **./8** in **Misc Build Option** in **Project Properties->Build** page.
- Fixed bug that project fails to run/debug after changing project SDK from 0.19.0 to 0.20.0.
- Fixed bug that **Misc Build Options** are not honored for s64 build.

Magic Leap PackageManager

ALL PACKAGES

Shing Cheng

### Lessons and Tutorial Projects

### Lumin Runtime Packages

- Lumin Animation Studio NOT INSTALLED
- Lumin Runtime Documentation 0.20.0 PENDING INSTALL
- Lumin Runtime Editor NOT INSTALLED
- Magic Script NOT INSTALLED
- Lumin Runtime SDK 0.20.0 PENDING INSTALL

### Unity® Packages

### Change List

Package	Version	Size	Status
Lumin Runtime SDK	0.20.0	0.19 GB	✓
Lumin Runtime Documentation	0.20.0	11 MB	✓

APPLY CANCEL

### Lumin Runtime Documentation (0.20.0)

Available Versions

Open the Lumin Runtime documentation in a browser

Open

### Lumin Runtime API Documentation Version 0.20.0

Magic Leap PackageManager

ALL PACKAGES

Common Packages

Lessons and Tutorial Projects

Lumin Runtime Packages

Unity® Packages

- Unity® API Documentation 0.20.0
- Unity® Editor
- Magic Leap Unity® Package 0.20.0

Unreal Packages

Change List

Package Name	Version	Size	Status
Unity® API Documentation	0.20.0	2.8 MB	✓
Magic Leap Unity® Package	0.20.0	0.14 GB	✓

APPLY CANCEL

Magic Leap Unity® Package (0.20.0)

Available Versions

Open folder with the Unity® package in a file browser

Open Folder

DEPENDENCIES

- Lumin SDK version 0.20.0

Unity® Package 0.20.0

- Lumin SDK Version 0.20.0
- Unity® Version: 2019.1.0b8
- Lumin OS 0.95

Magic Leap's hardware and software is in a qualified developer release state right now. Some features or functionality of the Magic Leap hardware or software may not be available, may contain bugs or other defects, and/or may experience delays or failures. Please take a close look at this document since it contains valuable guidance on the current state of Magic Leap's hardware and software to help developers make the best possible use of the platform during our pre-release period. Want to know more? Head to the forums on our developer portal for more guidance.

Breaking Changes

- Steps to follow to start working with Lumin (using our MLPlugin) on 2019.1:
  - Import the latest Legacy Input Helpers package (community.xr.legacy.inputhelpers) from Unity Package Manager. This is required in order to use our Unity package.
  - Enable Lumin development by turning on "Virtual Reality Supported" in Lumin/Player Settings/XR Settings and adding "Lumin SDK" to the list. This will import the Magic Leap XR package which includes core functionality like Graphics, Lifecycle, Meshing, etc for the Lumin platform.
- AR/Magic Leap/Camera (Magic Leap Camera) component required on Camera object for all scenes. All Magic Leap specific camera features are now contained in this component.
- XR/TrackedPoseDriver component required on Camera object for all scenes. Change the default settings to the following:
  - Device: Generic XR Device
  - Pose Source: Head
  - Tracking Type: Rotation and Position

## Step 2: Download Unity & Required Components

Recommendations - for production purposes do not use beta version

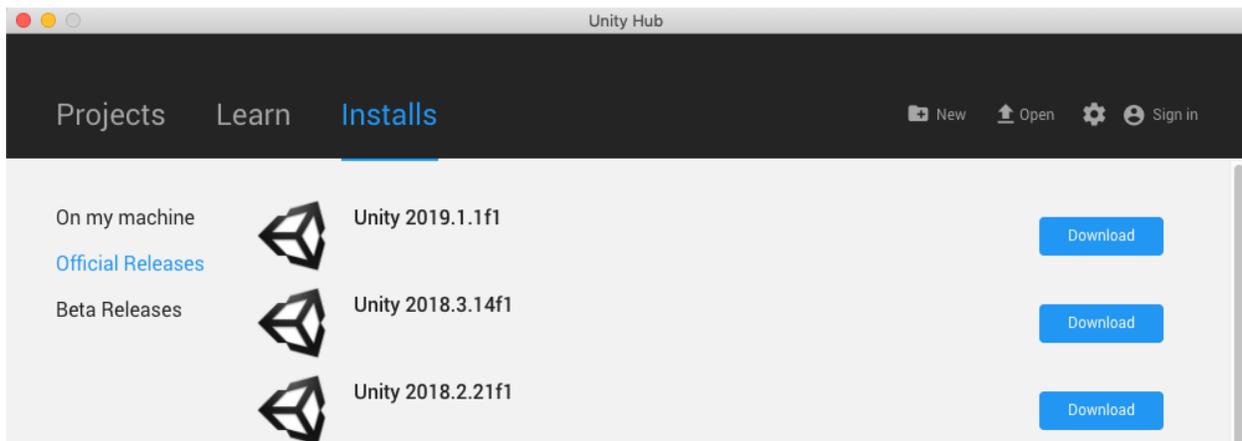
### Download Unity Hub

Go to <https://store.unity.com/download> and download the Unity Hub. The Unity Hub is a standalone application that streamlines the way you find, download, and manage your Unity Projects and installations. Follow standard download processes for your machine.

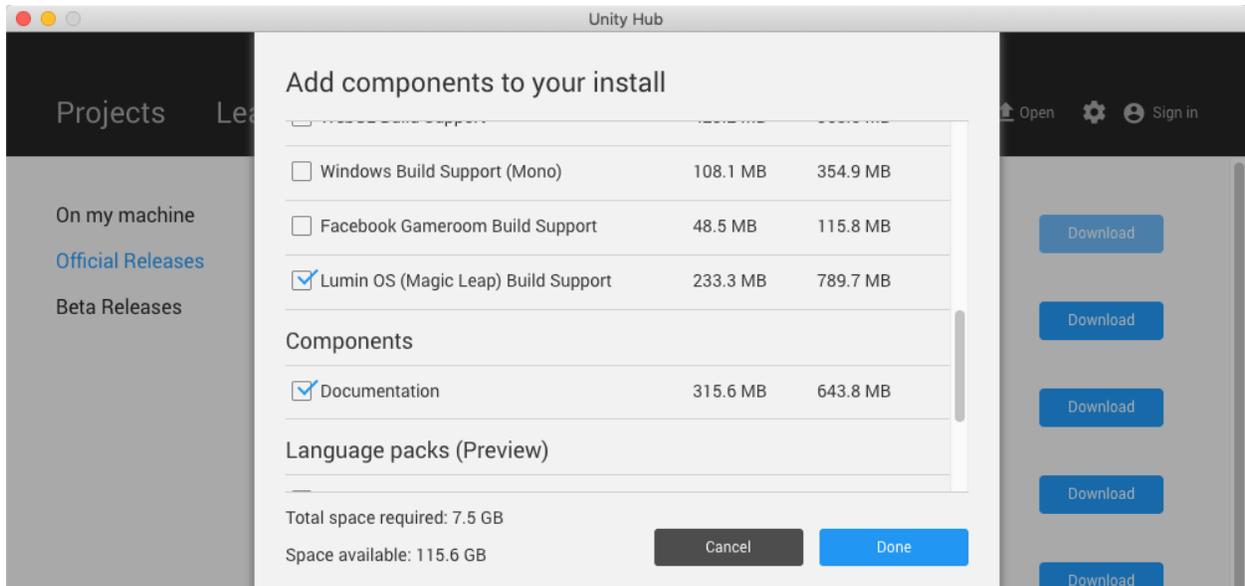
### Download Unity 2019 Version and Components

#### For Mac:

Once installed, under 'Installs,' 'Official Releases,' click 'Download' next to the recent Unity 2019 version (Not the beta version) - Beta versions are denoted with a **b** in its version name.



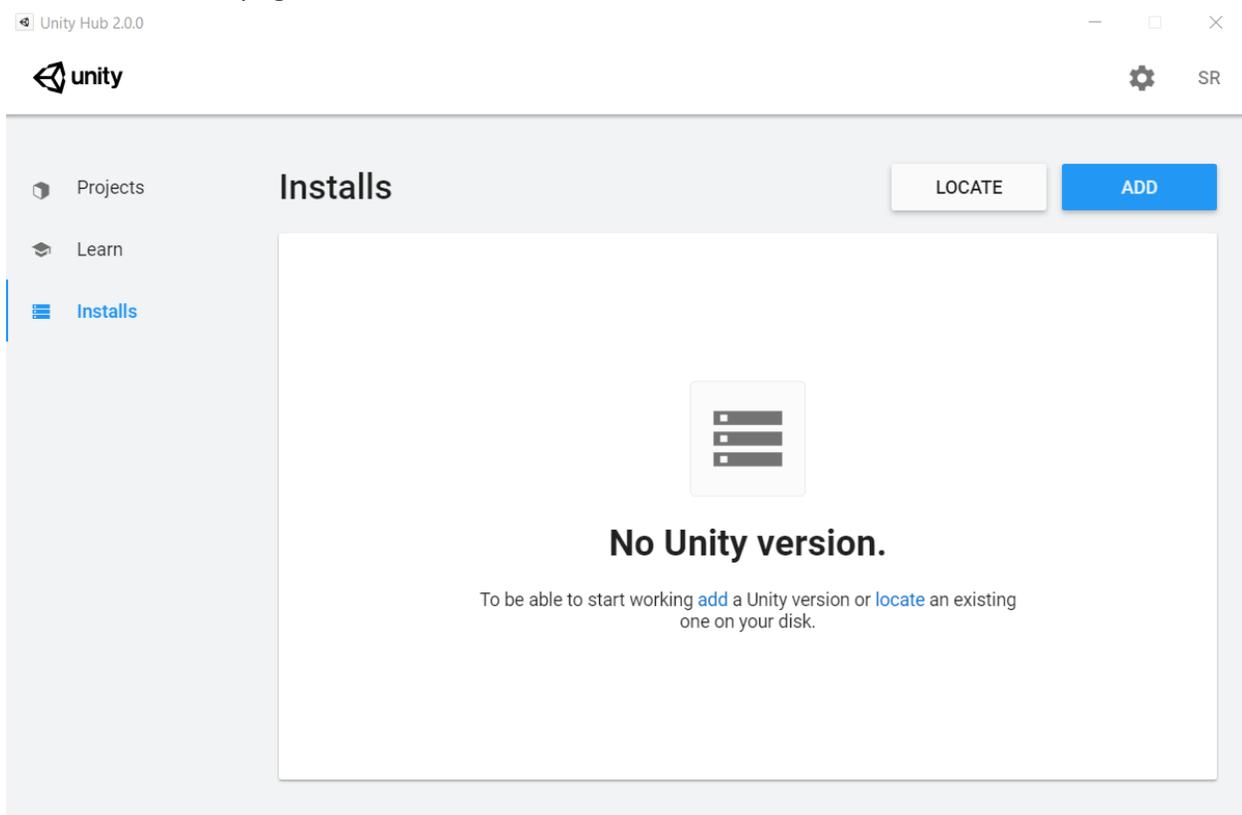
In the Add Components to your install dialog box, select the Lumin OS (Magic Leap) Build Support check box, and 'Documentation' check box, and then click Done.



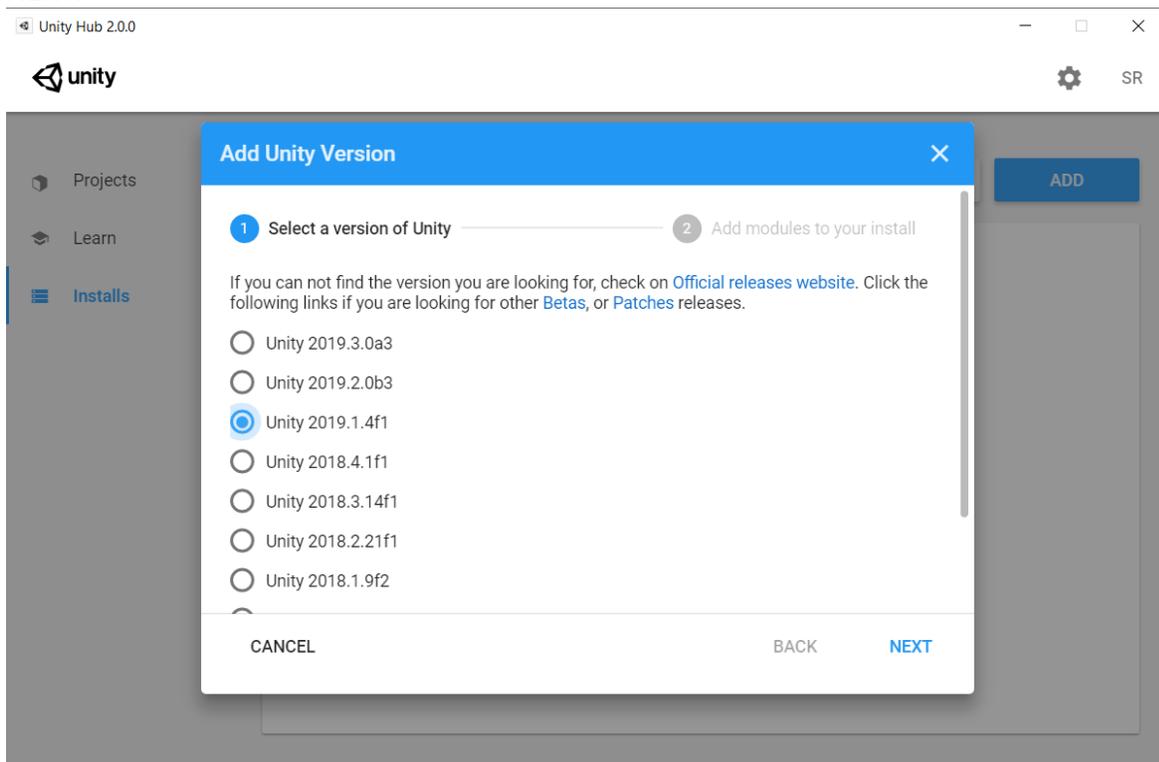
### For Windows:

When first opened, Unity Hub prompts the user for a license. Log in to Unity or create a Unity account and log in. Select “ACTIVATE NEW LICENSE” and create a new Unity Personal license.

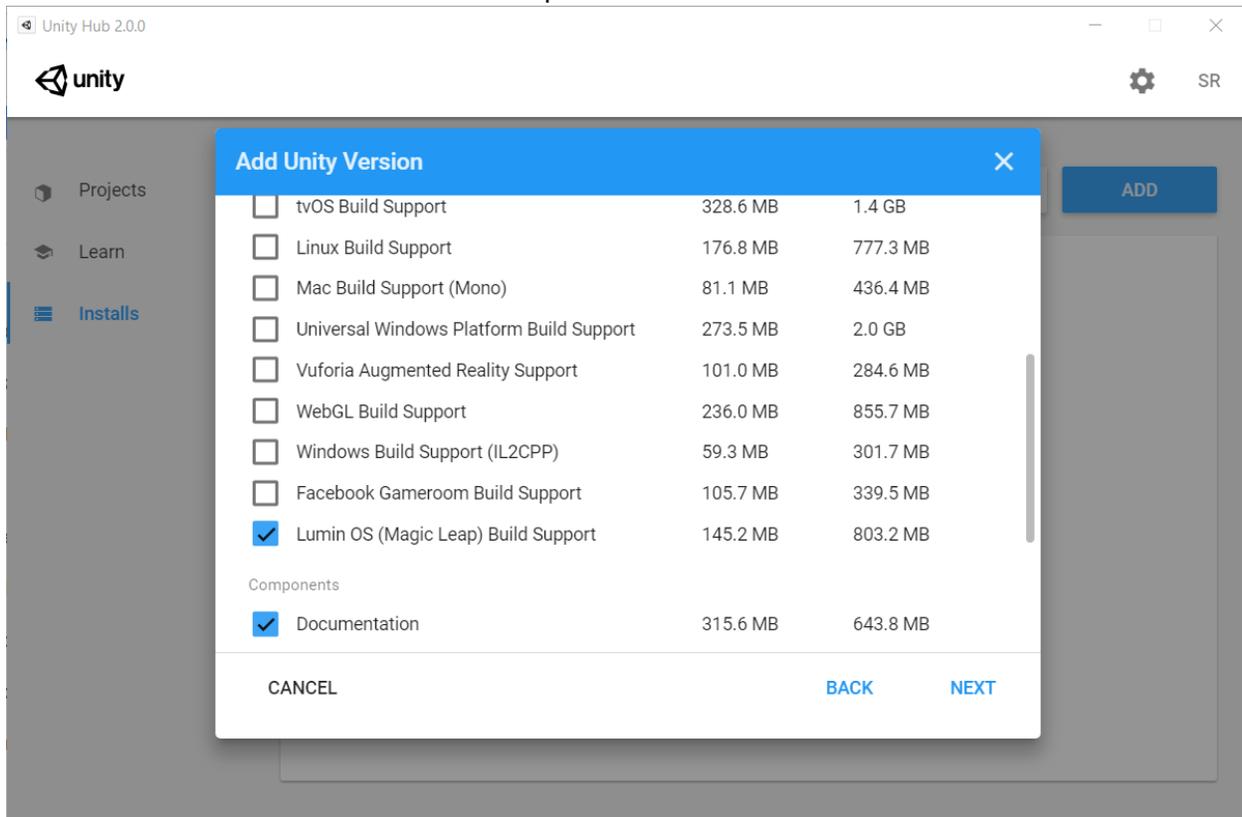
Go to the Installs page. Click ADD.



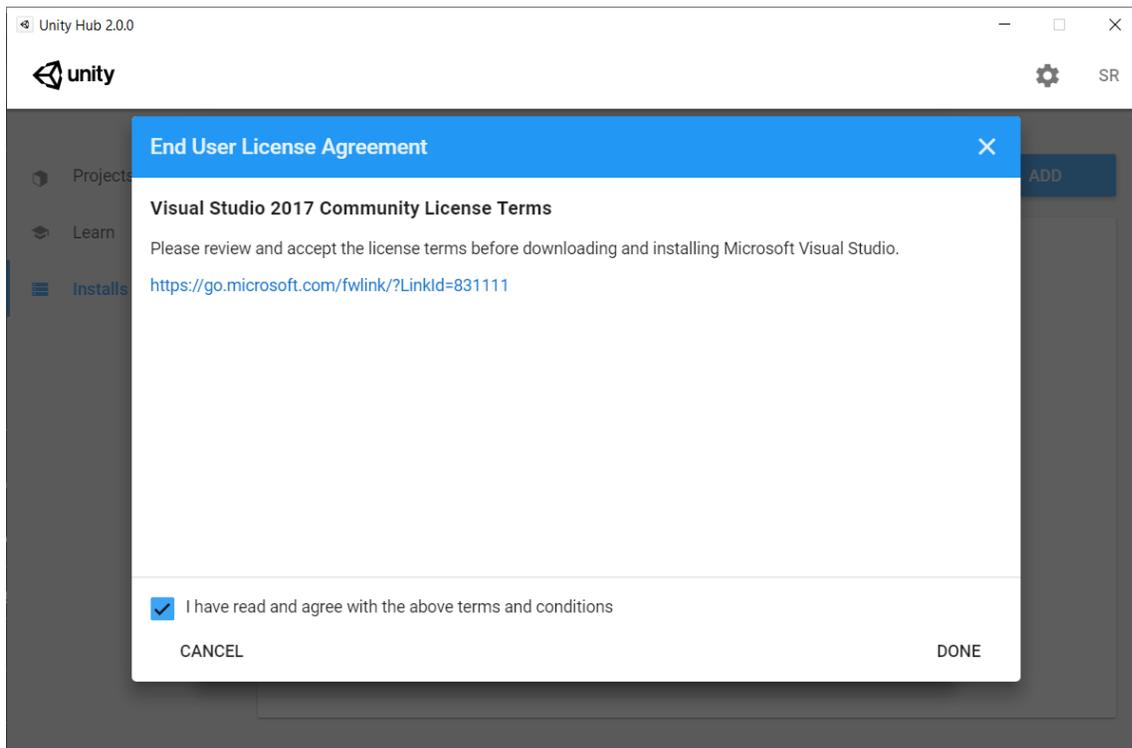
Select a 2019 version with "f" in the name, denoting that the version is a full version. Click NEXT.



Scroll down and select "Lumin OS" under platform. Click NEXT.



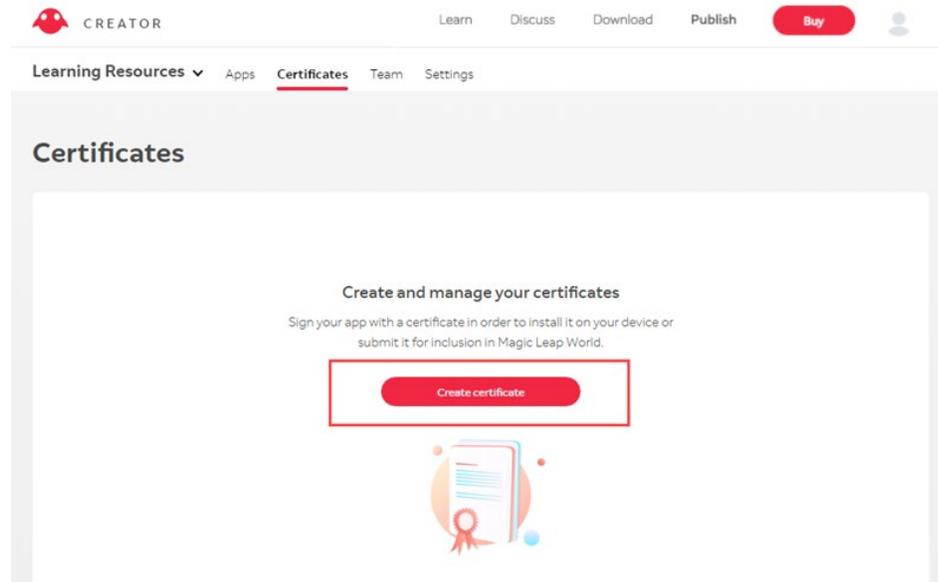
Agree to Visual Studio 2017 EULA and click DONE.



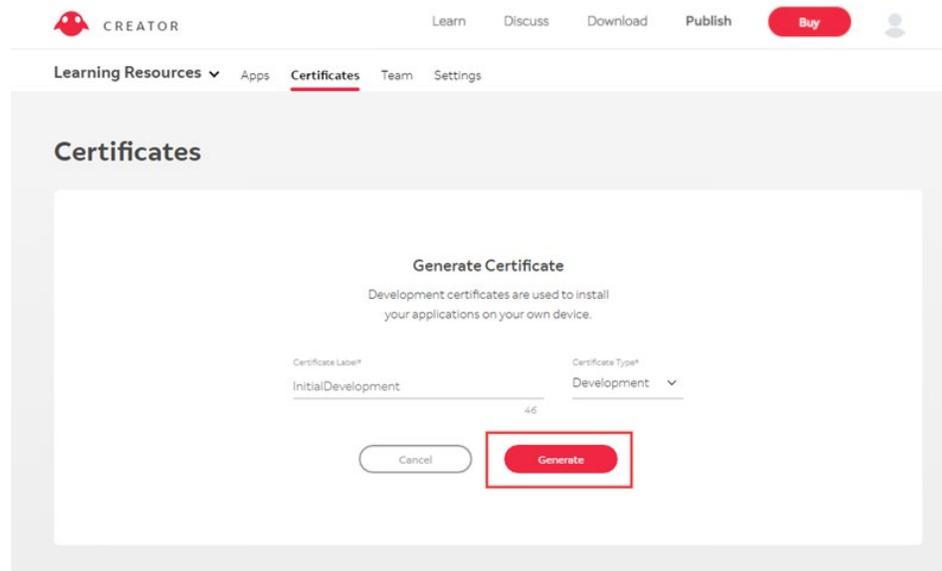
### Step 3: Create a Developer Certificate

Magic Leap Guide: <https://creator.magicleap.com/learn/guides/developer-certificates>

1. In the Magic Leap Creator Portal, click **Publish > Certificates**.
2. Click **Create certificate** or **Add new**.

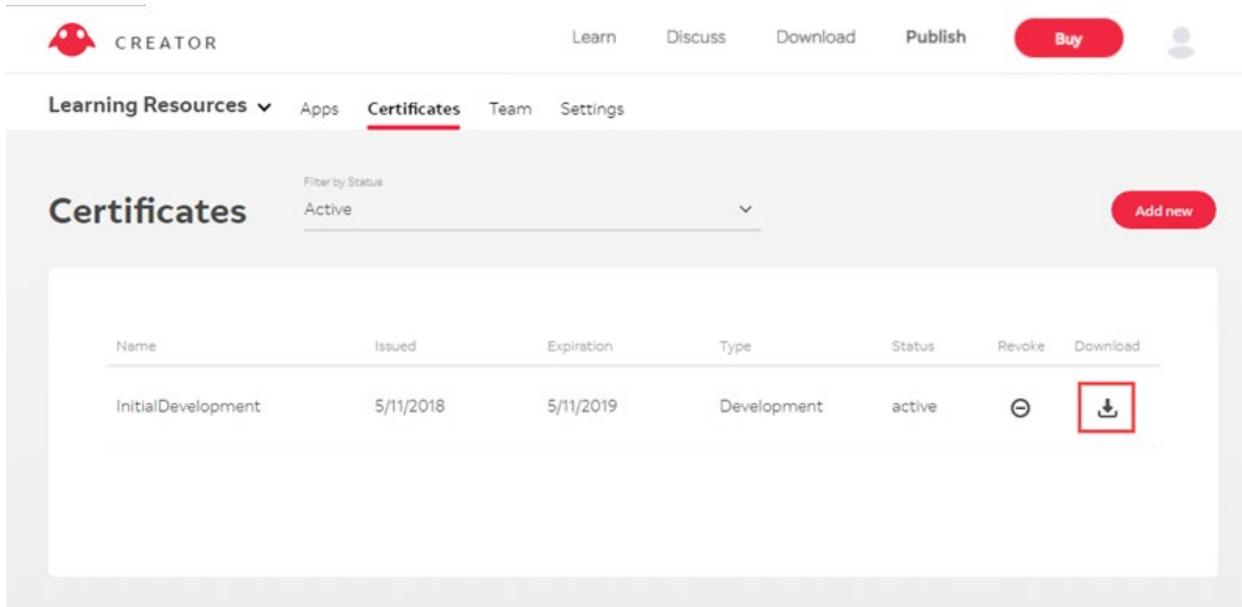


3. Enter a name for your Certificate. This is how you will differentiate this certificate from other certificates you generate. You can generate up to two at this time.



4. The privatekey.zip file will begin downloading. This is only available on this step. **If you misplace this file, you must generate a new certificate.**
5. The certificate takes a few minutes to generate. Refresh the page periodically until the certificate status changes from **pending** to **active**.

- When the certificate has generated, click the download button next to your certificate to download the .cert file.



- Extract the **.privkey** file.
- Move the **.privkey** and **.cert** files into the same folder. You can move them into your working directory.

After your **.privkey** and **.cert** files are downloaded and located in the same directory, you can [sign](#) your apps.

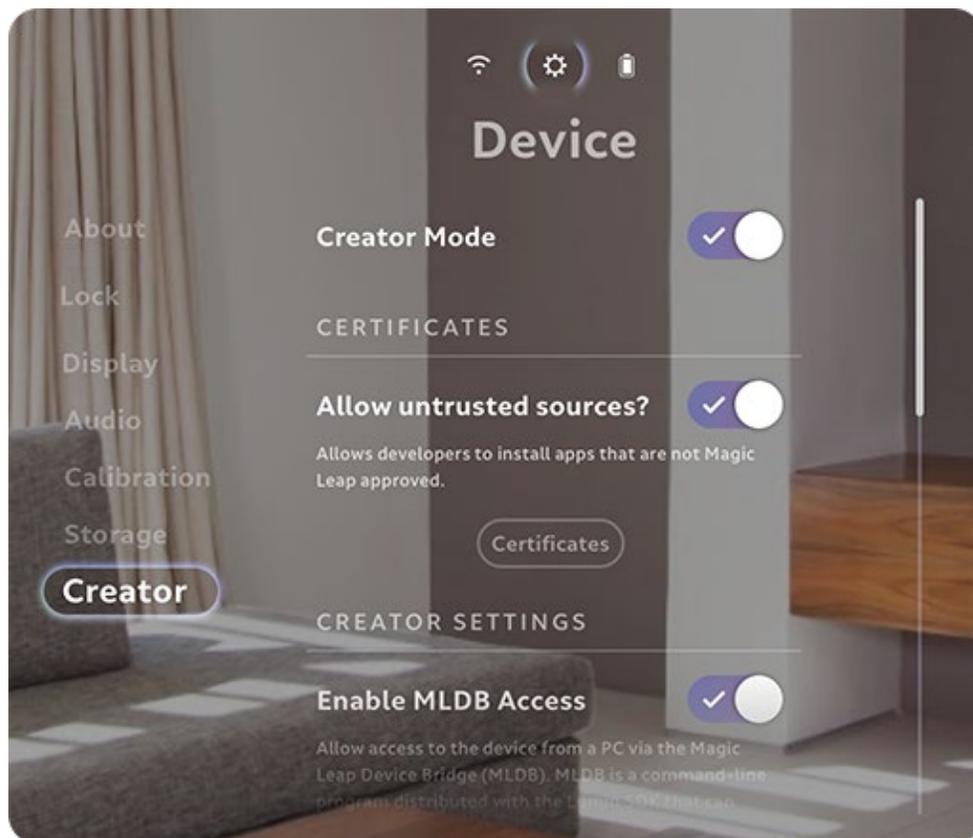
## Step 4: Set up Creator Mode in Magic Leap

<https://creator.magicleap.com/learn/guides/develop-device-setup>

You can use your Magic Leap One device for running, testing, and debugging your creations. But before you can do so, you must first enable developer settings on the device.

**Prerequisites: The following are REQUIRED for the menu to appear**

- An email address registered with Magic Leap on the [Creator Portal](#)
- An active Wi-Fi connection



To enable developer settings:

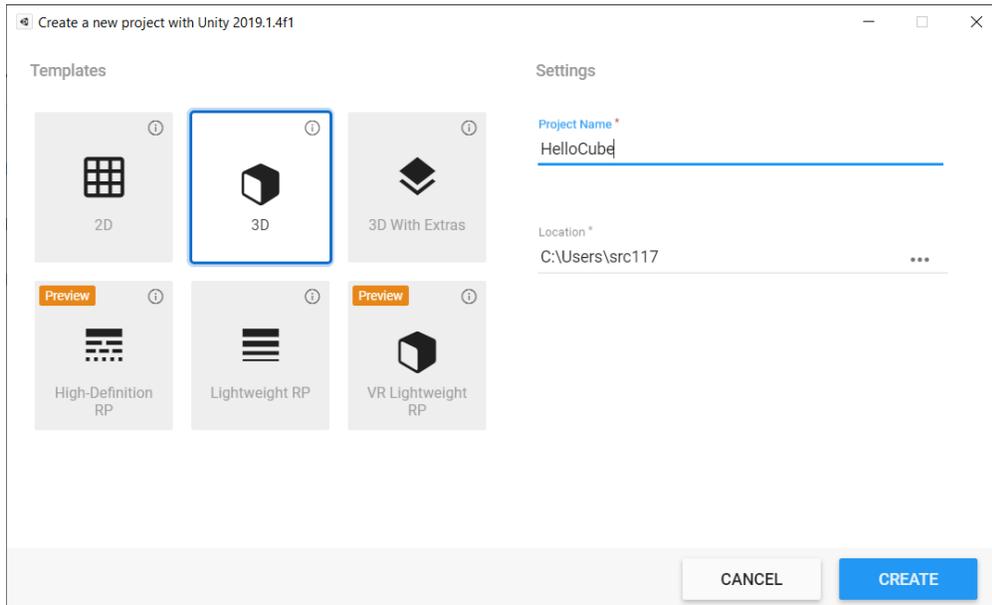
1. Put on the device and then turn it on.
2. If this is the first time the device has been turned on, follow the set up instructions you see in the Lightwear.
3. Using the Touchpad, go to the  icon above the Launcher, and then tap the Touchpad. The settings menu appears.
4. Go to Device, and then select Creator.

5. Note: If Creator isn't listed, make sure the device is connected to Wi-Fi, your email address is the same you registered with on the Creator Portal, and that you accepted the EULA when you registered your email address.
  6. Enable the following settings:
    - Creator Mode: Shows the Creator settings on the device.
    - Allow untrusted sources?: Lets the device load apps that are not Magic Leap approved. Note: Apps must still be signed with a certificate from the Creator Portal.
    - Enable MLDB Access: Lets the device communicate with a computer over USB or WiFi using Magic Leap Debug Bridge (MLDB)
    - Press Home on the Control twice to return to the Launcher.
-

## Step 5: Creating a Unity Project

### Part 1: Setting up the Environment

Open Unity, create a new project, '3D'. [Note - this takes time to import assets]



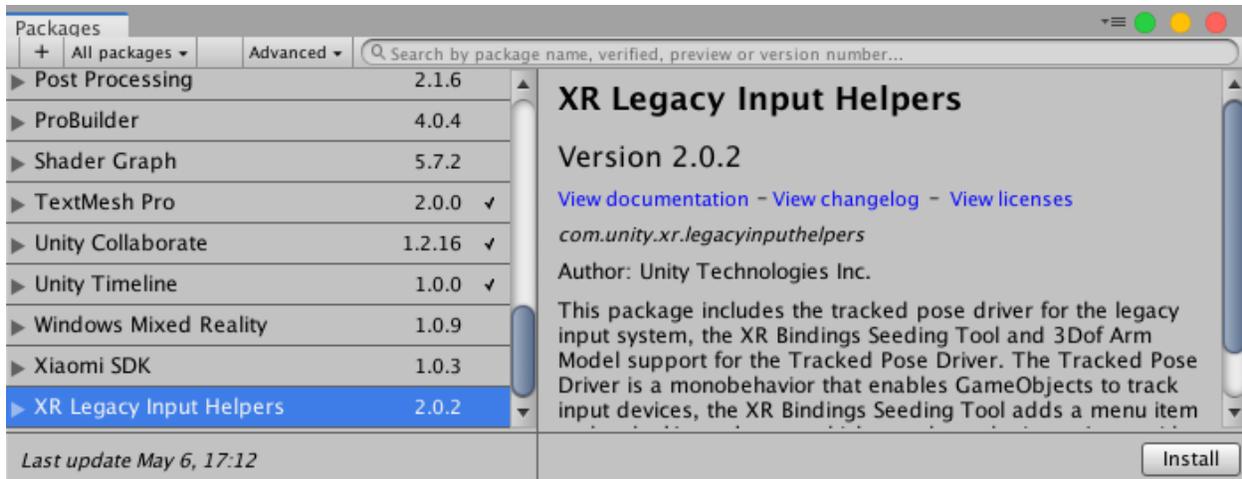
Customize layouts: - 2 by 3 layout is used in this tutorial

## Setting Preferences for Magic Leap Compatibility

### Install the Unity XR Legacy Input Helpers

In the Unity Editor main menu, go to **Window > Package Manager**. The Packages dialog box opens.

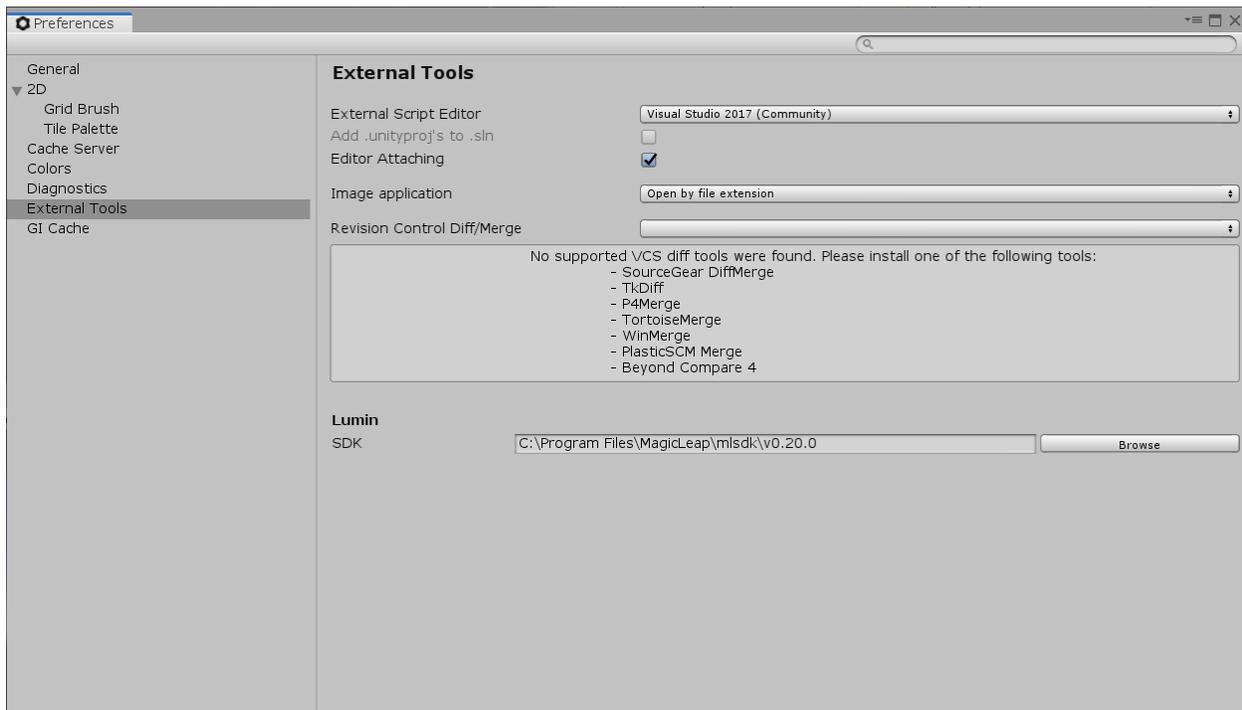
From the Packages list, select XR Legacy Input Helpers. Click Install.



## Set the Path to the Lumin SDK

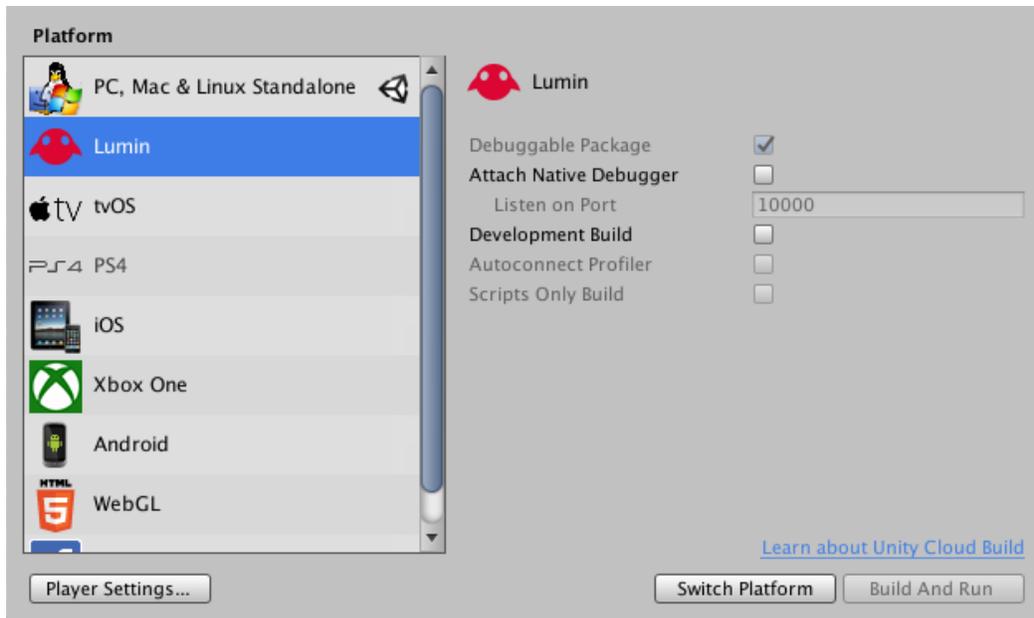
Go to **Edit>Preferences**(Windows) or **Unity>Preferences**(Mac)

In the sidebar, choose **External Tools** and select **Browse** in the **Lumin** section. Select the Lumin SDK folder (usually `/Users/user/MagicLeap/mlsdk/v0.xx.0`).



## Switch the Build Settings platform to Lumin

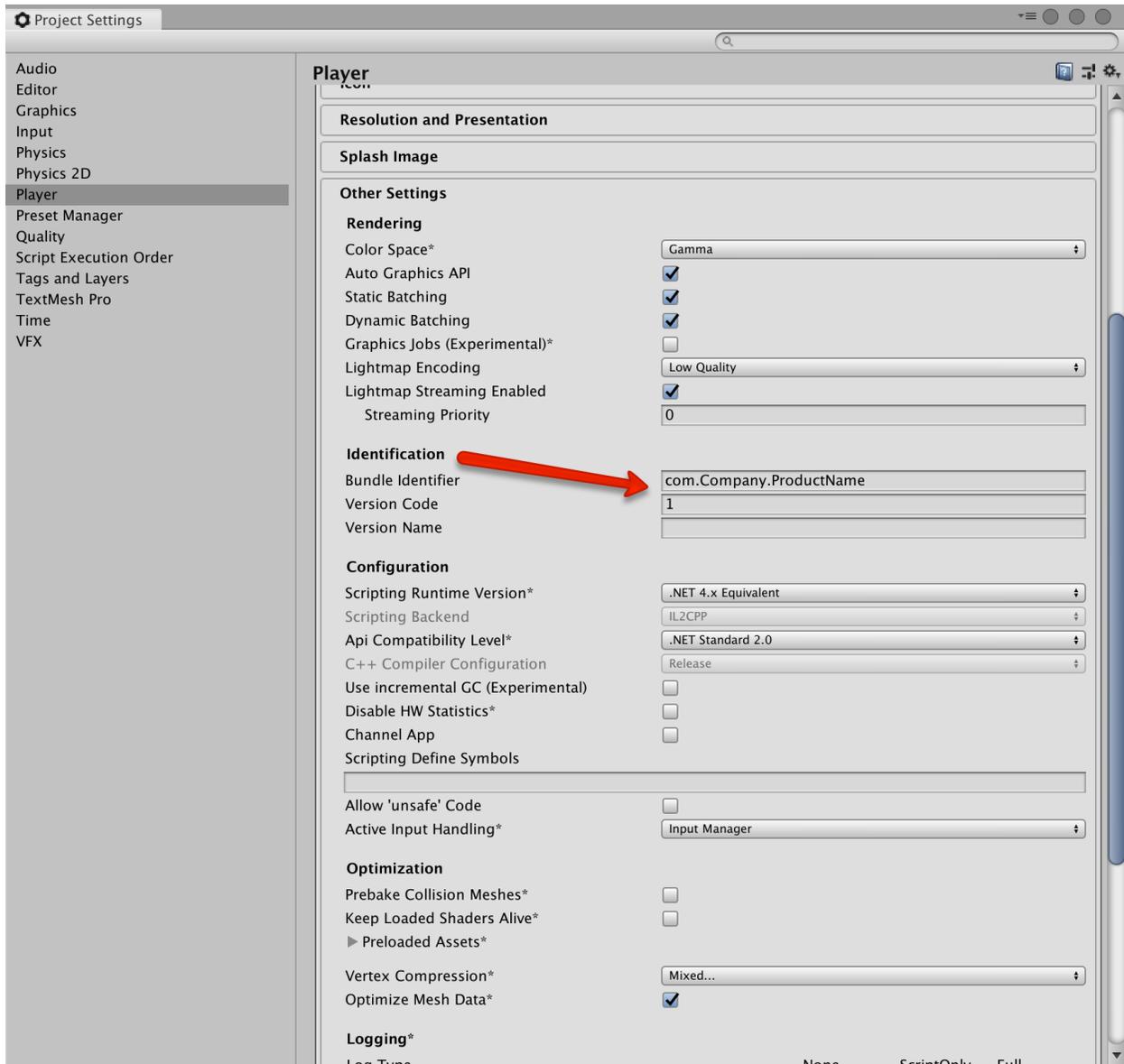
Go to **File > Build Settings**. From the Platform list, select **Lumin** and then click **Switch Platform**.



## Set the required Player settings for Lumin

Go to **Edit > Project Settings**. Select **Player**. Click the  tab.

- In **Other Settings**, change **Color Space** to **Linear**.
- In **Publishing Settings** select your certificate. Make sure the private key file is in the same folder as the certificate.
- In **XR Settings**, select the **Virtual Reality Supported** check box. In **XR Settings**, change **Stereo Rendering Mode** to **Single Pass Instanced**.



**NOTE:** Change the bundle identifier to “com.<companyname>.<appname>” all lowercase with no spaces. For instance: the app called “Hello Cube” produced by “UMIT” should have the identifier “com.umat.hellocube”.

## Import the Magic Leap Unity Package

1. Go to **Assets > Import Package**, and then click **Custom Package**.
2. In the **Import package** dialog box, select the MagicLeap.unitypackage file. It is typically located in Users/user/MagicLeap/tools/unity/v0.xx.0.
3. In the **Import Unity Package** dialog box, click **All** and then click **Import**.

## Use the Magic Leap prefab Main Camera

1. In the Hierarchy window, delete Main Camera from SampleScene.
2. Navigate to **Assets>MagicLeap>Core Components**. Drag **Main Camera** to SampleScene in the Hierarchy window.

### **Copy manifest.xml from Examples folder to Assets folder**

1. Navigate to **Assets>MagicLeap>Examples>Plugins>Lumin**.
2. Right-click **manifest**, and then click **Show in Explorer** (Windows) or **Reveal in Finder** (Mac).
3. Copy the **manifest.xml** file to the project's **Assets/Plugins/Lumin** folder – create this folder if it doesn't already exist.

## Part 2: Hello Cube

1. Under **Assets**, create a folder called **Materials**.
2. Right-click inside the folder and select **Create>Material** to create a new material and name it red.
3. Under Inspector, select **Albedo**, and click on the color field to select a color for the material. Make it red.
4. Navigate to **GameObject>3D Object>Cube** to create a cube.
5. In the **Inspector**, change its position to (0,2,2) and its scale to (0.5, 0.5, 0.5). To color the cube, drag the material from the materials folder to the cube's materials in the **Inspector** window.
6. In the **Scene Window**, make sure that the **Main Camera** is facing the cube. The cube will be visible in the **Game Window**.
7. Turn the ML1 headset on, and connect it to the computer.
8. To load the app onto the ML1 headset, go to **File>Build and Run**.
9. The app should launch automatically once it is loaded onto the ML1 headset.
10. If the **Main Camera** was positioned correctly, then the red cube should be visible. Otherwise try looking around for the cube. If the cube still can't be found, move the **Main Camera** again, and rebuild and load the app.